

# Decision Trees

---

Nipun Batra and teaching staff

July 21, 2025

IIT Gandhinagar

## **Discrete Input Discrete Output**

---

# The need for interpretability

## How to maintain trust in AI

Beyond developing initial trust, however, creators of AI also must work to maintain that trust. Siau and Wang suggest seven ways of "developing continuous trust" beyond the initial phases of product development:

- Usability and reliability. AI "should be designed to operate easily and intuitively," Siau and Wang write. "There should be no unexpected downtime or crashes."
- Collaboration and communication. AI developers want to create systems that perform autonomously, without human involvement. Developers must focus on creating AI applications that smoothly and easily collaborate and communicate with humans.
- Sociability and bonding. Building social activities into AI applications is one way to strengthen trust. A robotic dog that can recognize its owner and show affection is one example, Siau and Wang write.
- Security and privacy protection. AI applications rely on large data sets, so ensuring privacy and security will be crucial to establishing trust in the applications.
- Interpretability. Just as transparency is instrumental in building initial trust, interpretability – or the ability for a machine to explain its conclusions or actions – will help sustain trust.

## Training Data

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

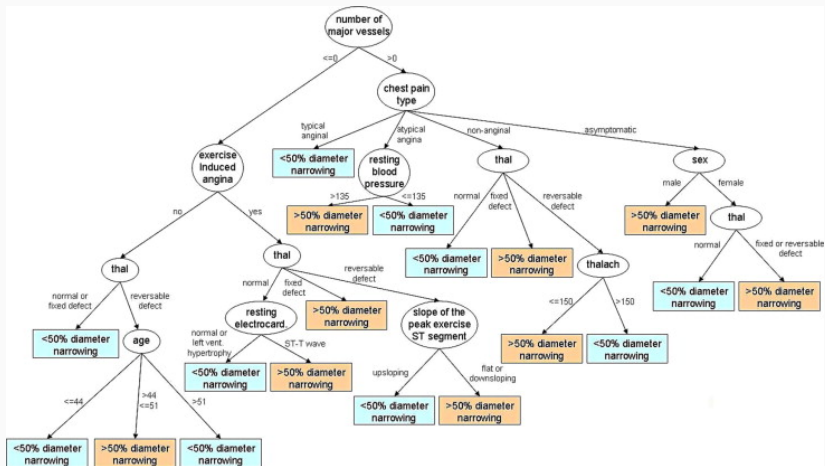
# Learning a Complicated Neural Network



# Learnt Decision Tree



# Medical Diagnosis using Decision Trees



Source: Improving medical decision trees by combining relevant health-care criteria



## Leo Breiman 1928-2005

Professor of Statistics, [UC Berkeley](#)  
Verified email at [stat.berkeley.edu](mailto:stat.berkeley.edu) - [Homepage](#)  
[Data Analysis](#) [Statistics](#) [Machine Learning](#)

[FOLLOW](#)

TITLE	CITED BY	YEAR
<a href="#">Random forests</a> L Breiman Machine learning 45 (1), 5-32	53816	2001
<a href="#">Classification and Regression Trees</a> L Breiman, JH Friedman, RA Olshen, CJ Stone CRC Press, New York	43992 *	1999
<a href="#">Classification and regression trees</a> L Breiman Chapman & Hall/CRC	43992 *	1984
<a href="#">Bagging predictors</a> L Breiman Machine learning 24 (2), 123-140	22742	1996
<a href="#">Statistical Modeling: The Two Cultures</a> L Breiman	2788 *	2003
<a href="#">Statistical modeling: The two cultures (with comments and a rejoinder by the author)</a> L Breiman Statistical Science 16 (3), 199-231	2772	2001
<a href="#">Estimating optimal transformations for multiple regression and correlation</a>	2086	1985

Cited by

[VIEW ALL](#)

	All	Since 2015
Citations	142857	68736
h-index	51	33
i10-index	80	46





Volume 5, number 1

INFORMATION PROCESSING LETTERS

May 1976

## CONSTRUCTING OPTIMAL BINARY DECISION TREES IS NP-COMPLETE\*

Laurent HYAFIL

*IRIA – Laboria, 78150 Rocquencourt, France*

and

Ronald L. RIVEST

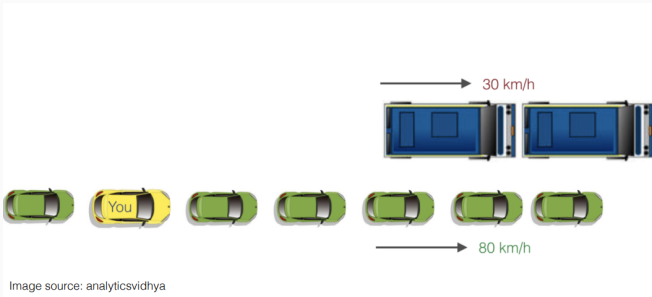
*Dept. of Electrical Engineering and Computer Science, M.I.T., Cambridge, Massachusetts 02139, USA*

Received 7 November 1975, revised version received 26 January 1976

Binary decision trees, computational complexity, NP-complete

# Greedy Algorithm

Core idea: At each level, choose an attribute that gives **biggest estimated** performance gain!



Greedy!≠Optimal

## Towards biggest estimated performance gain

## Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

## Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

## Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

# Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- For examples, we have 9 Yes, 5 No

# Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- For examples, we have 9 Yes, 5 No



# Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- For examples, we have 9 Yes, 5 No
- Would it be trivial if we had 14 Yes or 14 No?

# Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- For examples, we have 9 Yes, 5 No
- Would it be trivial if we had 14 Yes or 14 No?

# Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- For examples, we have 9 Yes, 5 No
- Would it be trivial if we had 14 Yes or 14 No?
- Yes!

# Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- For examples, we have 9 Yes, 5 No
- Would it be trivial if we had 14 Yes or 14 No?
- Yes!

# Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- For examples, we have 9 Yes, 5 No
- Would it be trivial if we had 14 Yes or 14 No?
- Yes!
- Key insight: Problem is “easier” when there is less disagreement

# Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- For examples, we have 9 Yes, 5 No
- Would it be trivial if we had 14 Yes or 14 No?
- Yes!
- Key insight: Problem is “easier” when there is less disagreement

# Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- For examples, we have 9 Yes, 5 No
- Would it be trivial if we had 14 Yes or 14 No?
- Yes!
- Key insight: Problem is “easier” when there is less disagreement
- Need some statistical measure of “disagreement”

# Entropy

Statistical measure to characterize the (im)purity of examples

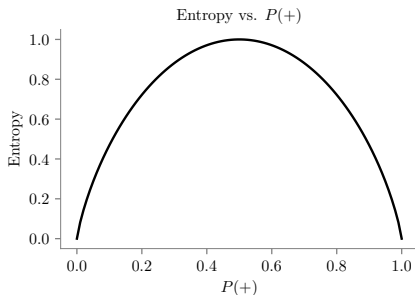


# Entropy

Statistical measure to characterize the (im)purity of examples

$$H(X) = - \sum_{i=1}^k p(x_i) \log_2 p(x_i)$$

Notebook: [entropy.html](#)



## Towards biggest estimated performance gain

## Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

## Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

# Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

# Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- Can we use Outlook as the root node?

# Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- Can we use Outlook as the root node?

# Towards biggest estimated performance gain

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- Can we use Outlook as the root node?
- When Outlook is overcast, we always Play and thus no “disagreement”



Reduction in entropy by partitioning examples ( $S$ ) on attribute  $A$

$$\text{Gain}(S, A) \equiv \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

## ID3 (Examples, Target Attribute, Attributes)

- Create a root node for tree

## ID3 (Examples, Target Attribute, Attributes)

- Create a root node for tree
- If all examples are  $+/-$ , return root with label =  $+/-$

## ID3 (Examples, Target Attribute, Attributes)

- Create a root node for tree
- If all examples are  $+/-$ , return root with label =  $+/-$
- If attributes = empty, return root with most common value of Target Attribute in Examples

## ID3 (Examples, Target Attribute, Attributes)

- Create a root node for tree
- If all examples are  $+/-$ , return root with label =  $+/-$
- If attributes = empty, return root with most common value of Target Attribute in Examples
- Begin

## ID3 (Examples, Target Attribute, Attributes)

- Create a root node for tree
- If all examples are  $+/-$ , return root with label =  $+/-$
- If attributes = empty, return root with most common value of Target Attribute in Examples
- Begin
  - $A \leftarrow$  attribute from Attributes which best classifies Examples

## ID3 (Examples, Target Attribute, Attributes)

- Create a root node for tree
- If all examples are  $+/-$ , return root with label =  $+/-$
- If attributes = empty, return root with most common value of Target Attribute in Examples
- Begin
  - $A \leftarrow$  attribute from Attributes which best classifies Examples
  - $\text{Root} \leftarrow A$

## ID3 (Examples, Target Attribute, Attributes)

- Create a root node for tree
- If all examples are  $+/-$ , return root with label =  $+/-$
- If attributes = empty, return root with most common value of Target Attribute in Examples
- Begin
  - $A \leftarrow$  attribute from Attributes which best classifies Examples
  - Root  $\leftarrow A$
  - For each value ( $v$ ) of  $A$



## ID3 (Examples, Target Attribute, Attributes)

- Create a root node for tree
- If all examples are  $+/-$ , return root with label =  $+/-$
- If attributes = empty, return root with most common value of Target Attribute in Examples
- Begin
  - $A \leftarrow$  attribute from Attributes which best classifies Examples
  - $\text{Root} \leftarrow A$
  - For each value ( $v$ ) of  $A$ 
    - Add new tree branch :  $A = v$

## ID3 (Examples, Target Attribute, Attributes)

- Create a root node for tree
- If all examples are  $+/-$ , return root with label =  $+/-$
- If attributes = empty, return root with most common value of Target Attribute in Examples
- Begin
  - $A \leftarrow$  attribute from Attributes which best classifies Examples
  - $\text{Root} \leftarrow A$
  - For each value ( $v$ ) of  $A$ 
    - Add new tree branch :  $A = v$
    - $\text{Examples}_v$ : subset of examples that  $A = v$

## ID3 (Examples, Target Attribute, Attributes)

- Create a root node for tree
- If all examples are  $+/-$ , return root with label =  $+/-$
- If attributes = empty, return root with most common value of Target Attribute in Examples
- Begin
  - $A \leftarrow$  attribute from Attributes which best classifies Examples
  - $\text{Root} \leftarrow A$
  - For each value ( $v$ ) of  $A$ 
    - Add new tree branch :  $A = v$
    - $\text{Examples}_v$ : subset of examples that  $A = v$
    - If  $\text{Examples}_v$  is empty: add leaf with label = most common value of Target Attribute

## ID3 (Examples, Target Attribute, Attributes)

- Create a root node for tree
- If all examples are  $+/-$ , return root with label =  $+/-$
- If attributes = empty, return root with most common value of Target Attribute in Examples
- Begin
  - $A \leftarrow$  attribute from Attributes which best classifies Examples
  - $\text{Root} \leftarrow A$
  - For each value ( $v$ ) of  $A$ 
    - Add new tree branch :  $A = v$
    - $\text{Examples}_v$ : subset of examples that  $A = v$
    - If  $\text{Examples}_v$  is empty: add leaf with label = most common value of Target Attribute
    - Else: ID3 ( $\text{Examples}_v$ , Target attribute, Attributes -  $A$ )

# Learnt Decision Tree

Root Node (empty)

## Training Data

Day	Outlook	Temp	Humidity	Windy	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

## Entropy calculated

We have 14 examples in  $S$ : 5 No, 9 Yes

$$\begin{aligned} (S) &= -p_{\text{No}} \log_2 p_{\text{No}} - p_{\text{Yes}} \log_2 p_{\text{Yes}} \\ &= -\frac{5}{14} \log_2 \left( \frac{5}{14} \right) - \frac{9}{14} \log_2 \left( \frac{9}{14} \right) = 0.940 \end{aligned}$$

## Information Gain for Outlook

Outlook	Play
Sunny	No
Sunny	No
Overcast	Yes
Rain	Yes
Rain	Yes
Rain	No
Overcast	Yes
Sunny	No
Sunny	Yes
Rain	Yes
Sunny	Yes
Overcast	Yes
Overcast	Yes
Rain	No





## Information Gain for Outlook

Outlook	Play
Sunny	No
Sunny	No
Sunny	No
Sunny	Yes
Sunny	Yes

We have 2 Yes, 3 No

$$= -\frac{3}{5} \log_2 \left( \frac{3}{5} \right) - \frac{2}{5} \log_2 \left( \frac{2}{5} \right) = 0.971$$

## Information Gain for Outlook

Outlook	Play
Sunny	No
Sunny	No
Sunny	No
Sunny	Yes
Sunny	Yes

We have 2 Yes, 3 No

$$= -\frac{3}{5} \log_2 \left( \frac{3}{5} \right) - \frac{2}{5} \log_2 \left( \frac{2}{5} \right) = 0.971$$

## Information Gain for Outlook

Outlook	Play
Sunny	No
Sunny	No
Sunny	No
Sunny	Yes
Sunny	Yes

We have 2 Yes, 3 No

$$\begin{aligned} &= -\frac{3}{5} \log_2 \left( \frac{3}{5} \right) - \\ &\frac{2}{5} \log_2 \left( \frac{2}{5} \right) = 0.971 \end{aligned}$$

Outlook	Play
Overcast	Yes
Overcast	Yes
Overcast	Yes
Overcast	Yes

We have 4 Yes, 0 No

$$= 0 \text{ (pure subset)}$$

## Information Gain for Outlook

Outlook	Play
Sunny	No
Sunny	No
Sunny	No
Sunny	Yes
Sunny	Yes

We have 2 Yes, 3 No

$$\begin{aligned} &= -\frac{3}{5} \log_2 \left( \frac{3}{5} \right) - \\ &\frac{2}{5} \log_2 \left( \frac{2}{5} \right) = 0.971 \end{aligned}$$

Outlook	Play
Overcast	Yes
Overcast	Yes
Overcast	Yes
Overcast	Yes

We have 4 Yes, 0 No

$$= 0 \text{ (pure subset)}$$

## Information Gain for Outlook

Outlook	Play
Sunny	No
Sunny	No
Sunny	No
Sunny	Yes
Sunny	Yes

We have 2 Yes, 3 No

$$= -\frac{3}{5} \log_2 \left( \frac{3}{5} \right) - \frac{2}{5} \log_2 \left( \frac{2}{5} \right) = 0.971$$

Outlook	Play
Overcast	Yes
Overcast	Yes
Overcast	Yes
Overcast	Yes

We have 4 Yes, 0 No

$$= 0 \text{ (pure subset)}$$

Outlook	Play
Rain	Yes
Rain	Yes
Rain	No
Rain	Yes
Rain	No

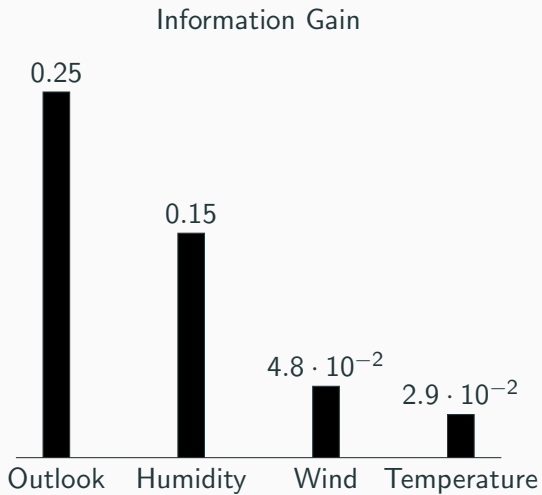
We have 3 Yes, 2 No

$$= -\frac{3}{5} \log_2 \left( \frac{3}{5} \right) - \frac{2}{5} \log_2 \left( \frac{2}{5} \right) = 0.971$$

$$\text{Gain}(S, \text{Outlook}) = \text{Entropy}(S) - \sum_{v \in \{\text{Rain}, \text{Sunny}, \text{Overcast}\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

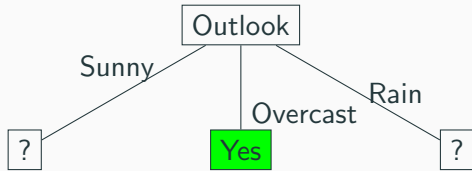
$$\begin{aligned} \text{Gain}(S, \text{Outlook}) &= \text{Entropy}(S) - \frac{5}{14} \text{Entropy}(S_{\text{Sunny}}) - \frac{4}{14} \text{Entropy}(S_{\text{Overcast}}) - \frac{5}{14} \text{Entropy}(S_{\text{Rain}}) \\ &= 0.940 - \frac{5}{14} \times 0.971 - \frac{4}{14} \times 0 - \frac{5}{14} \times 0.971 = 0.940 - 0.347 - 0 - 0.347 = 0.246 \end{aligned}$$

# Information Gain





# Learnt Decision Tree



## Calling ID3 on Outlook=Sunny

Day	Temp	Humidity	Windy	Play
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

## Calling ID3 on Outlook=Sunny

Day	Temp	Humidity	Windy	Play
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

- Gain( $S_{\text{Outlook=Sunny}}$ , Temp) = Entropy(2 Yes, 3 No) -  
(2/5)\*Entropy(0 Yes, 2 No) -(2/5)\*Entropy(1 Yes, 1 No) -  
(1/5)\*Entropy(1 Yes, 0 No)

## Calling ID3 on Outlook=Sunny

Day	Temp	Humidity	Windy	Play
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

- Gain( $S_{\text{Outlook=Sunny}}$ , Temp) = Entropy(2 Yes, 3 No) -  
(2/5)\*Entropy(0 Yes, 2 No) -(2/5)\*Entropy(1 Yes, 1 No) -  
(1/5)\*Entropy(1 Yes, 0 No)

## Calling ID3 on Outlook=Sunny

Day	Temp	Humidity	Windy	Play
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

- $\text{Gain}(S_{\text{Outlook}=\text{Sunny}}, \text{Temp}) = \text{Entropy}(2 \text{ Yes}, 3 \text{ No}) - (2/5) * \text{Entropy}(0 \text{ Yes}, 2 \text{ No}) - (2/5) * \text{Entropy}(1 \text{ Yes}, 1 \text{ No}) - (1/5) * \text{Entropy}(1 \text{ Yes}, 0 \text{ No})$
- $\text{Gain}(S_{\text{Outlook}=\text{Sunny}}, \text{Humidity}) = \text{Entropy}(2 \text{ Yes}, 3 \text{ No}) - (2/5) * \text{Entropy}(2 \text{ Yes}, 0 \text{ No}) - (3/5) * \text{Entropy}(0 \text{ Yes}, 3 \text{ No})$   
 $\implies$  **maximum possible for the set**

## Calling ID3 on Outlook=Sunny

Day	Temp	Humidity	Windy	Play
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

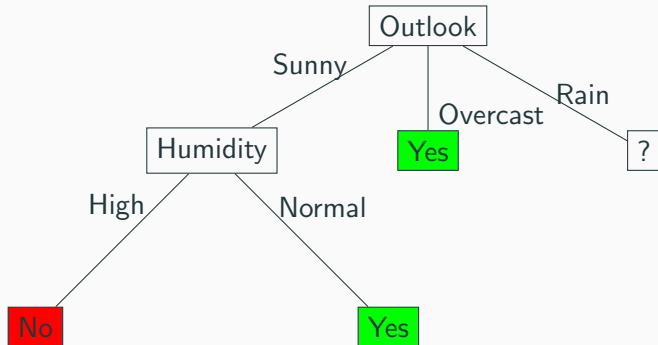
- $\text{Gain}(S_{\text{Outlook}=\text{Sunny}}, \text{Temp}) = \text{Entropy}(2 \text{ Yes}, 3 \text{ No}) - (2/5) * \text{Entropy}(0 \text{ Yes}, 2 \text{ No}) - (2/5) * \text{Entropy}(1 \text{ Yes}, 1 \text{ No}) - (1/5) * \text{Entropy}(1 \text{ Yes}, 0 \text{ No})$
- $\text{Gain}(S_{\text{Outlook}=\text{Sunny}}, \text{Humidity}) = \text{Entropy}(2 \text{ Yes}, 3 \text{ No}) - (2/5) * \text{Entropy}(2 \text{ Yes}, 0 \text{ No}) - (3/5) * \text{Entropy}(0 \text{ Yes}, 3 \text{ No})$   
 $\implies$  **maximum possible for the set**

## Calling ID3 on Outlook=Sunny

Day	Temp	Humidity	Windy	Play
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

- $\text{Gain}(S_{\text{Outlook}=\text{Sunny}}, \text{Temp}) = \text{Entropy}(2 \text{ Yes}, 3 \text{ No}) - (2/5) * \text{Entropy}(0 \text{ Yes}, 2 \text{ No}) - (2/5) * \text{Entropy}(1 \text{ Yes}, 1 \text{ No}) - (1/5) * \text{Entropy}(1 \text{ Yes}, 0 \text{ No})$
- $\text{Gain}(S_{\text{Outlook}=\text{Sunny}}, \text{Humidity}) = \text{Entropy}(2 \text{ Yes}, 3 \text{ No}) - (2/5) * \text{Entropy}(2 \text{ Yes}, 0 \text{ No}) - (3/5) * \text{Entropy}(0 \text{ Yes}, 3 \text{ No})$   
 $\implies$  **maximum possible for the set**
- $\text{Gain}(S_{\text{Outlook}=\text{Sunny}}, \text{Windy}) = \text{Entropy}(2 \text{ Yes}, 3 \text{ No}) - (3/5) * \text{Entropy}(1 \text{ Yes}, 2 \text{ No}) - (2/5) * \text{Entropy}(1 \text{ Yes}, 1 \text{ No})$

# Learnt Decision Tree





## Calling ID3 on (Outlook=Rain)

Day	Temp	Humidity	Windy	Play
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

- The attribute Windy gives the highest information gain

# Learnt Decision Tree



# Prediction for Decision Tree

Just walk down the tree!



## Prediction for Decision Tree

Just walk down the tree!



Prediction for <High Humidity, Strong Wind, Sunny Outlook, Hot Temp> is ?

## Prediction for Decision Tree

Just walk down the tree!



Prediction for <High Humidity, Strong Wind, Sunny Outlook, Hot Temp> is ?  
No

## Limiting Depth of Tree

Assuming if you were only allowed depth-1 trees, how would it look for the current dataset?

## Limiting Depth of Tree

Assuming if you were only allowed depth-1 trees, how would it look for the current dataset?

Apply the same rules, except when depth limit is reached, the leaf node is assigned the most common occurring value in that path.

## Limiting Depth of Tree

Assuming if you were only allowed depth-1 trees, how would it look for the current dataset?

Apply the same rules, except when depth limit is reached, the leaf node is assigned the most common occurring value in that path.

What is depth-0 tree (no decision) for the examples?



## Limiting Depth of Tree

Assuming if you were only allowed depth-1 trees, how would it look for the current dataset?

Apply the same rules, except when depth limit is reached, the leaf node is assigned the most common occurring value in that path.

What is depth-0 tree (no decision) for the examples?

Always predicting Yes

## Limiting Depth of Tree

Assuming if you were only allowed depth-1 trees, how would it look for the current dataset?

Apply the same rules, except when depth limit is reached, the leaf node is assigned the most common occurring value in that path.

What is depth-0 tree (no decision) for the examples?

Always predicting Yes

What is depth-1 tree (no decision) for the examples?

# Limiting Depth of Tree

Assuming if you were only allowed depth-1 trees, how would it look for the current dataset?

Apply the same rules, except when depth limit is reached, the leaf node is assigned the most common occurring value in that path.

What is depth-0 tree (no decision) for the examples?

Always predicting Yes

What is depth-1 tree (no decision) for the examples?



## Discrete Input, Real Output

---

## Modified Dataset

Day	Outlook	Temp	Humidity	Wind	Minutes Played
D1	Sunny	Hot	High	Weak	20
D2	Sunny	Hot	High	Strong	24
D3	Overcast	Hot	High	Weak	40
D4	Rain	Mild	High	Weak	50
D5	Rain	Cool	Normal	Weak	60
D6	Rain	Cool	Normal	Strong	10
D7	Overcast	Cool	Normal	Strong	4
D8	Sunny	Mild	High	Weak	10
D9	Sunny	Cool	Normal	Weak	60
D10	Rain	Mild	Normal	Weak	40
D11	Sunny	Mild	High	Strong	45
D12	Overcast	Mild	High	Strong	40
D13	Overcast	Hot	Normal	Weak	35
D14	Rain	Mild	High	Strong	20

## Measure of Impurity for Regression?

## Measure of Impurity for Regression?

## Measure of Impurity for Regression?

- Any guesses?



## Measure of Impurity for Regression?

- Any guesses?

## Measure of Impurity for Regression?

- Any guesses?
- Mean Squared Error

## Measure of Impurity for Regression?

- Any guesses?
- Mean Squared Error

## Measure of Impurity for Regression?

- Any guesses?
- Mean Squared Error
- $\text{MSE}(S) = 311.34$

## Measure of Impurity for Regression?

- Any guesses?
- Mean Squared Error
- $\text{MSE}(S) = 311.34$

## Measure of Impurity for Regression?

- Any guesses?
- Mean Squared Error
- $\text{MSE}(S) = 311.34$
- What about splitting criterion for regression?

## Measure of Impurity for Regression?

- Any guesses?
- Mean Squared Error
- $\text{MSE}(S) = 311.34$
- What about splitting criterion for regression?

## Measure of Impurity for Regression?

- Any guesses?
- Mean Squared Error
- $MSE(S) = 311.34$
- What about splitting criterion for regression?
- **MSE Reduction** (not Information Gain!)



## Measure of Impurity for Regression?

- Any guesses?
- Mean Squared Error
- $MSE(S) = 311.34$
- What about splitting criterion for regression?
- **MSE Reduction** (not Information Gain!)

## Measure of Impurity for Regression?

- Any guesses?
- Mean Squared Error
- $\text{MSE}(S) = 311.34$
- What about splitting criterion for regression?
- **MSE Reduction** (not Information Gain!)
- $\text{MSE Reduction} = \text{MSE}(S) - \sum_v \frac{|S_v|}{|S|} \text{MSE}(S_v)$

## Gain by splitting on Wind

## Gain by splitting on Wind

Wind	Minutes Played
Weak	20
Strong	24
Weak	40
Weak	50
Weak	60
Strong	10
Strong	4
Weak	10
Weak	60
Weak	40
Strong	45
Strong	40
Weak	35
Strong	20

$$\text{MSE}(S)=311.34$$

# Gain by splitting on Wind

Wind	Minutes Played
Weak	20
Strong	24
Weak	40
Weak	50
Weak	60
Strong	10
Strong	4
Weak	10
Weak	60
Weak	40
Strong	45
Strong	40
Weak	35
Strong	20

$$\text{MSE}(S) = 311.34$$

Wind	Minutes Played
Weak	20
Weak	40
Weak	50
Weak	60
Weak	10
Weak	60
Weak	40
Weak	35

$$\text{MSE}(S_{\text{Wind=Weak}}) = 277, \text{Weight} = \frac{8}{14}$$

Wind	Minutes Played
Strong	24
Strong	10
Strong	4
Strong	45
Strong	40
Strong	20

$$\text{MSE}(S_{\text{Wind=Strong}}) = 218, \text{Weight} = \frac{6}{14}$$

# MSE Reduction Calculation

**Correct calculation for Wind split:**

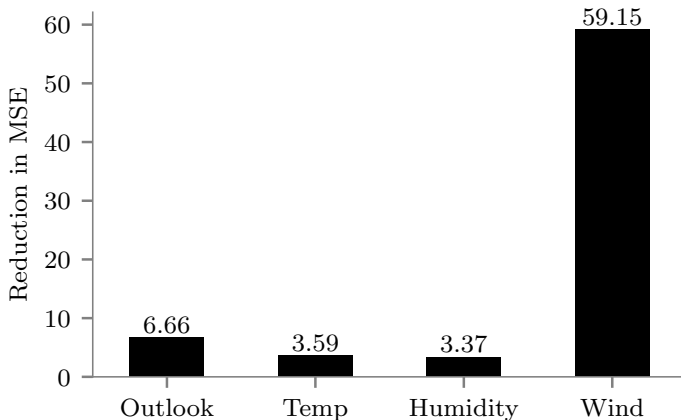
$$\begin{aligned}\text{MSE Reduction} &= \text{MSE}(S) - \text{Weighted Average MSE} \\ &= 311.34 - \left[ \frac{8}{14} \times 277 + \frac{6}{14} \times 218 \right] = 311.34 - [158.857 + 93.429] = 311.34 - 252.286 = 59.054\end{aligned}$$

**Key insight:** MSE Reduction  $> 0$  means the split improves our model!

**For regression:** Use MSE Reduction, NOT Information Gain!

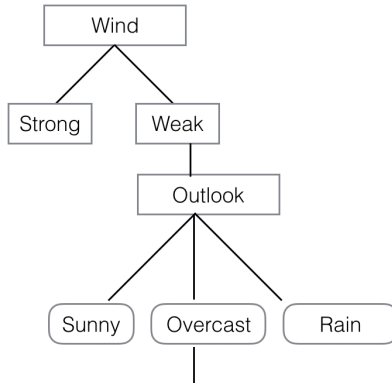
# MSE Reduction for Regression Trees

Notebook: [decision-tree-real-output.html](#)



# Learnt Tree

Assume a tree  
like this is  
learnt ...



	Day	Outlook	Temp	Humidity	Wind	Minutes Played
2	D3	Overcast	Hot	High	Weak	40
12	D13	Overcast	Hot	Normal	Weak	35



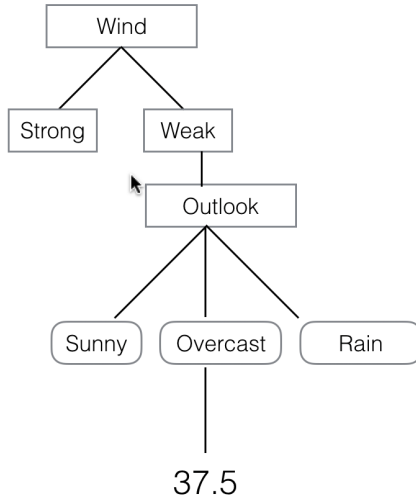
# Learnt Tree

## Method 1

Mins

Played=(40+35)

/2



## Real Input Discrete Output

---

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- How do you find splits?

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- How do you find splits?
- Sort by attribute

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- How do you find splits?
- Sort by attribute
- Find potential split points (midpoints).

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- How do you find splits?
- Sort by attribute
- Find potential split points (midpoints).
- For the above example, we have 5 potential splits: 44, 54, 66, 76, 85

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- How do you find splits?
- Sort by attribute
- Find potential split points (midpoints).
- For the above example, we have 5 potential splits: 44, 54, 66, 76, 85
- Calculate the weighted impurity for each split

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- How do you find splits?
- Sort by attribute
- Find potential split points (midpoints).
- For the above example, we have 5 potential splits: 44, 54, 66, 76, 85
- Calculate the weighted impurity for each split
- Choose the split with the lowest impurity



## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- Consider split at 44

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- Consider split at 44
- LHS has 1 No and 0 Yes; RHS has 3 Yes and 2 No

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- Consider split at 44
- LHS has 1 No and 0 Yes; RHS has 3 Yes and 2 No
- Entropy for LHS = 0, Entropy for RHS = 0.971

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- Consider split at 44
- LHS has 1 No and 0 Yes; RHS has 3 Yes and 2 No
- Entropy for LHS = 0, Entropy for RHS = 0.971
- Weighted Entropy =  $0.971 * 5/6 = 0.808$

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- Consider split at 54

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- Consider split at 54
- LHS has 2 No and 0 Yes; RHS has 3 Yes and 1 No

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- Consider split at 54
- LHS has 2 No and 0 Yes; RHS has 3 Yes and 1 No
- Entropy for LHS = 0, Entropy for RHS = 0.811

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- Consider split at 54
- LHS has 2 No and 0 Yes; RHS has 3 Yes and 1 No
- Entropy for LHS = 0, Entropy for RHS = 0.811
- Weighted Entropy =  $0.811 * 4/6 = 0.541$



## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- Consider split at 66

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- Consider split at 66
- LHS has 2 No and 1 Yes; RHS has 2 Yes and 1 No

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- Consider split at 66
- LHS has 2 No and 1 Yes; RHS has 2 Yes and 1 No
- Entropy for LHS = 0.918, Entropy for RHS = 0.918

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- Consider split at 66
- LHS has 2 No and 1 Yes; RHS has 2 Yes and 1 No
- Entropy for LHS = 0.918, Entropy for RHS = 0.918
- Weighted Entropy =  $0.918 \cdot 3/6 + 0.918 \cdot 3/6 = 0.918$

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- Consider split at 76

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- Consider split at 76
- LHS has 2 No and 2 Yes; RHS has 1 Yes and 1 No

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

- Consider split at 76
- LHS has 2 No and 2 Yes; RHS has 1 Yes and 1 No
- Entropy for LHS = 1, Entropy for RHS = 1

## Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

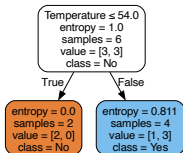
- Consider split at 76
- LHS has 2 No and 2 Yes; RHS has 1 Yes and 1 No
- Entropy for LHS = 1, Entropy for RHS = 1
- Weighted Entropy =  $1 \cdot 4/6 + 1 \cdot 2/6 = 1$



# Finding splits

Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

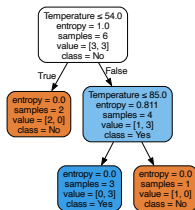
Notebook: [decision-tree-real-input-discrete-output.html](#)



# Finding splits

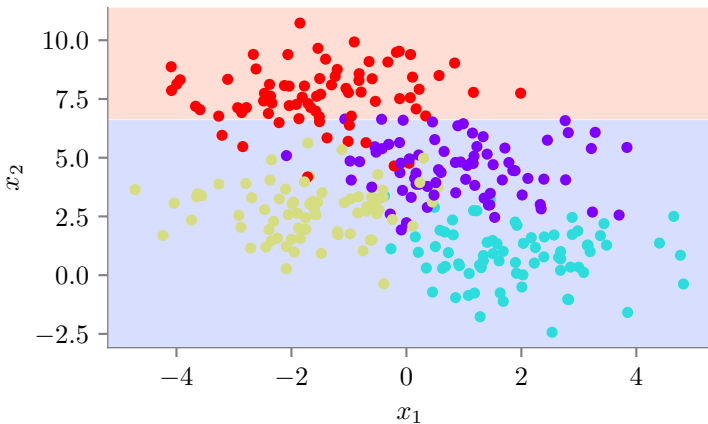
Day	Temperature	PlayTennis
D1	40	No
D2	48	No
D3	60	Yes
D4	72	Yes
D5	80	Yes
D6	90	No

Notebook: [decision-tree-real-input-discrete-output.html](#)



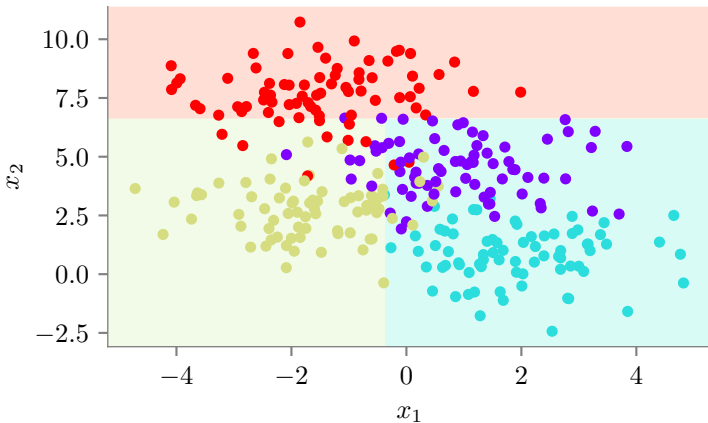
## Example (DT of depth 1)

Notebook: [decision-tree-real-input-discrete-output.html](#)



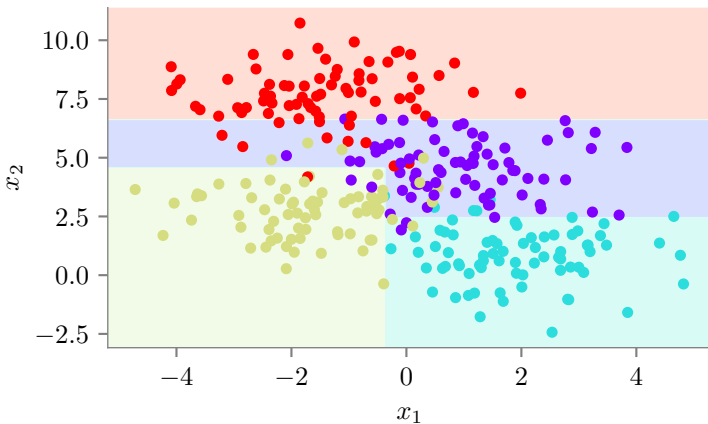
## Example (DT of depth 2)

Notebook: [decision-tree-real-input-discrete-output.html](#)



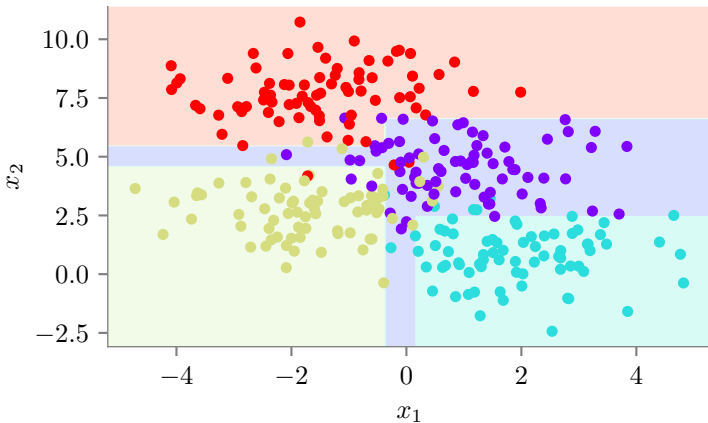
## Example (DT of depth 3)

Notebook: [decision-tree-real-input-discrete-output.html](#)



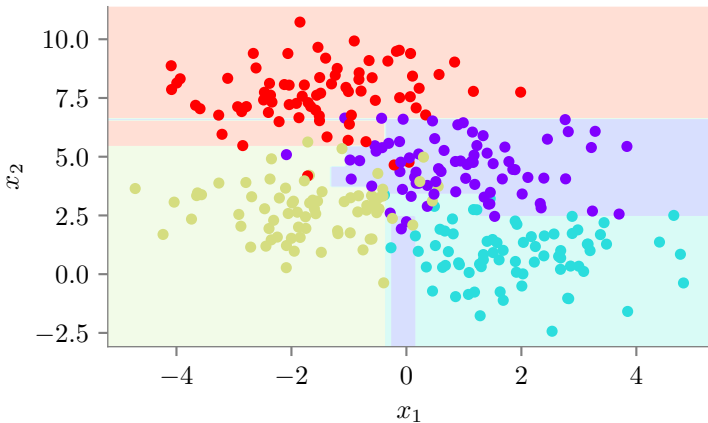
## Example (DT of depth 4)

Notebook: [decision-tree-real-input-discrete-output.html](#)



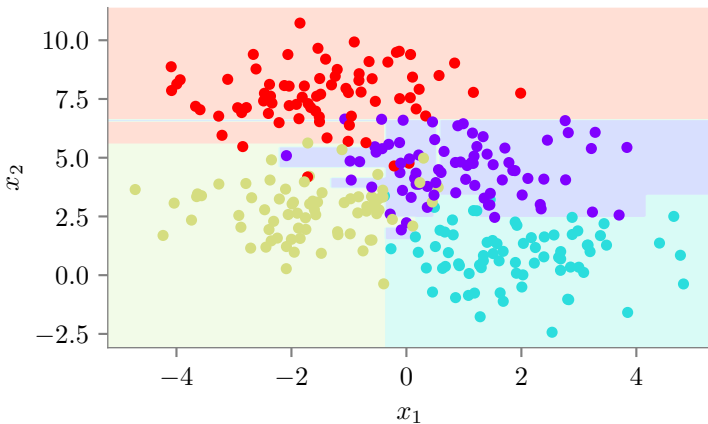
## Example (DT of depth 5)

Notebook: [decision-tree-real-input-discrete-output.html](#)



## Example (DT of depth 6)

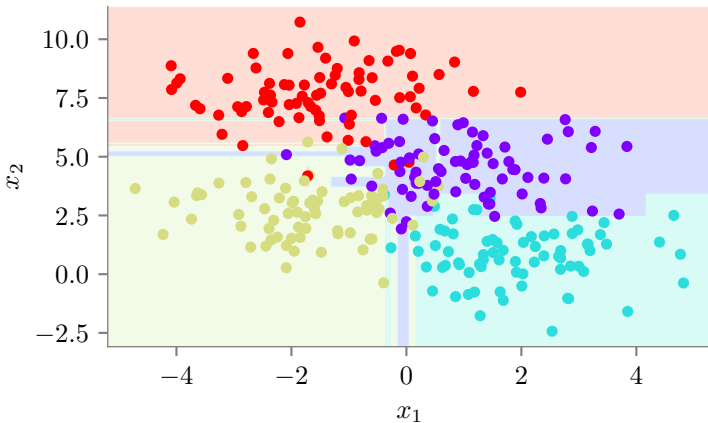
Notebook: [decision-tree-real-input-discrete-output.html](#)





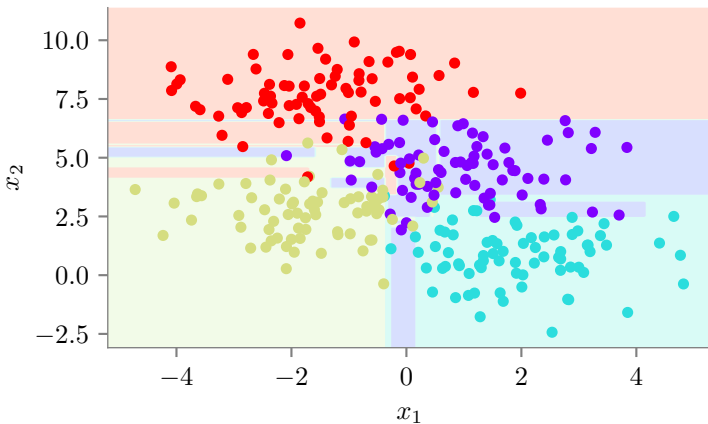
## Example (DT of depth 7)

Notebook: [decision-tree-real-input-discrete-output.html](#)



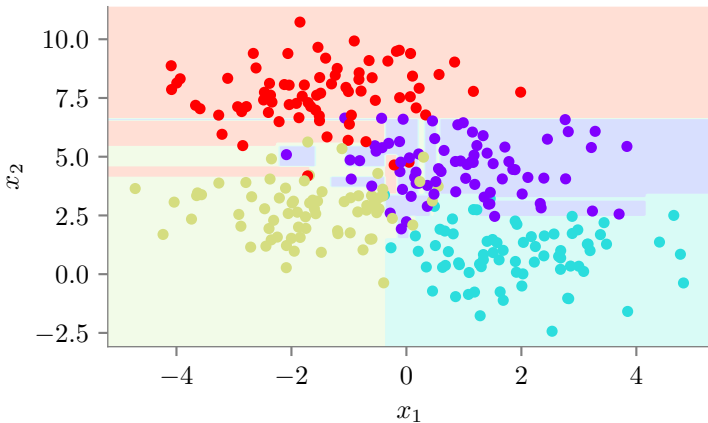
## Example (DT of depth 8)

Notebook: [decision-tree-real-input-discrete-output.html](#)



## Example (DT of depth 9)

Notebook: [decision-tree-real-input-discrete-output.html](#)



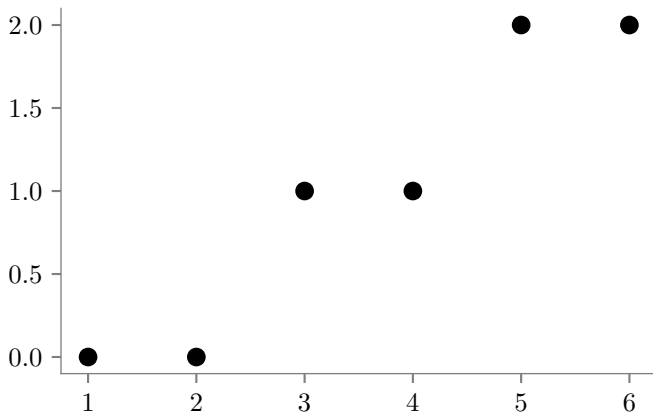
**Real Input Real Output**

---

## Example 1

Let us consider the dataset given below

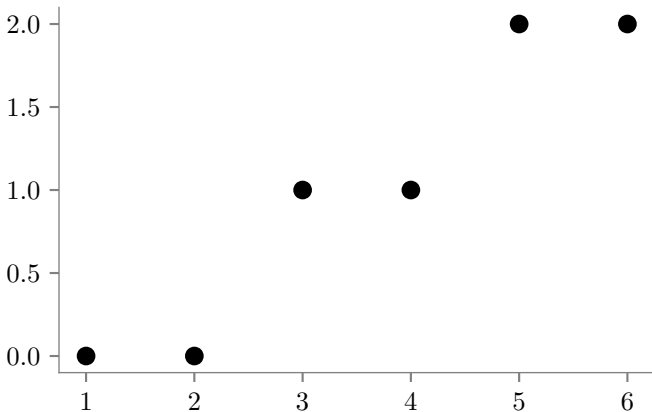
**Notebook:** [decision-tree-real-input-real-output.html](#)



## Example 1

What would be the prediction for decision tree with depth 0?

**Notebook:** [decision-tree-real-input-real-output.html](#)

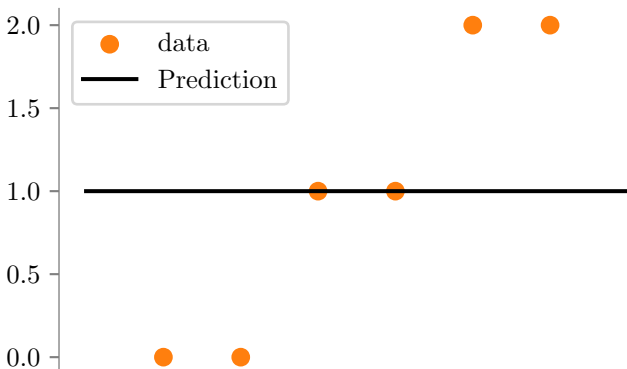


## Example 1

Prediction for decision tree with depth 0.

Horizontal dashed line shows the predicted  $Y$  value. It is the average of  $Y$  values of all datapoints.

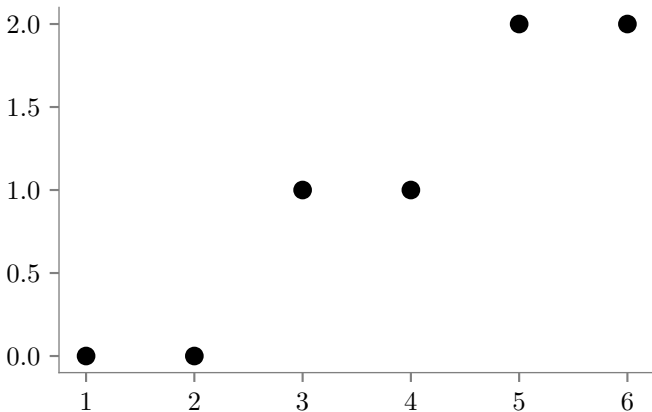
Notebook: [decision-tree-real-input-real-output.html](#)



## Example 1

What would be the decision tree with depth 1?

Notebook: [decision-tree-real-input-real-output.html](#)

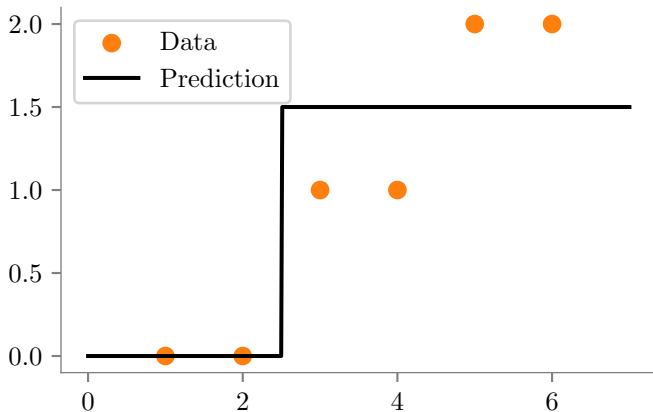




## Example 1

Decision tree with depth 1

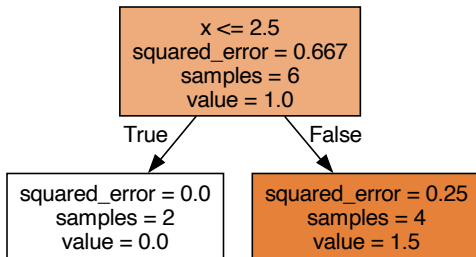
Notebook: [decision-tree-real-input-real-output.html](#)



# Example 1

## The Decision Boundary

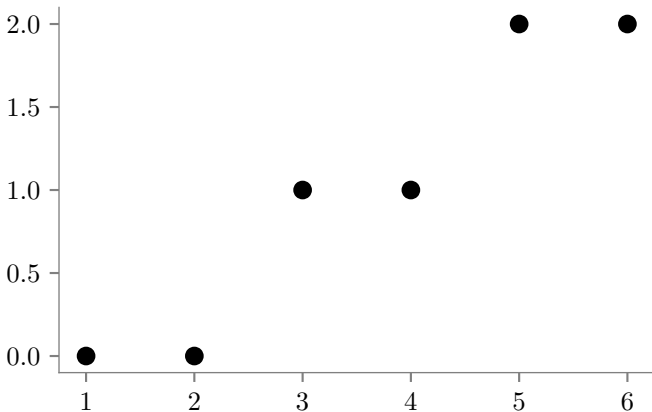
Notebook: [decision-tree-real-input-real-output.html](#)



## Example 1

What would be the decision tree with depth 2 ?

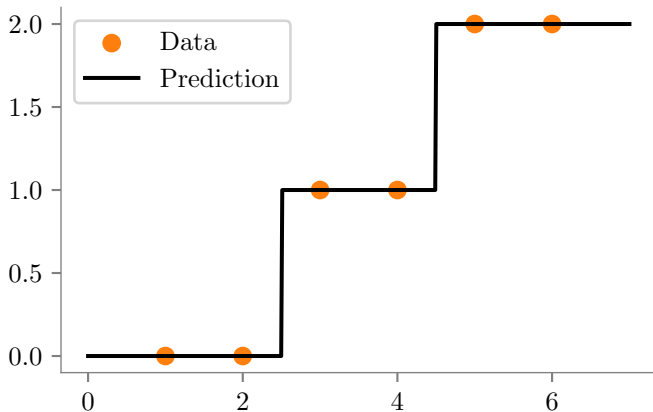
Notebook: [decision-tree-real-input-real-output.html](#)



## Example 1

Decision tree with depth 1

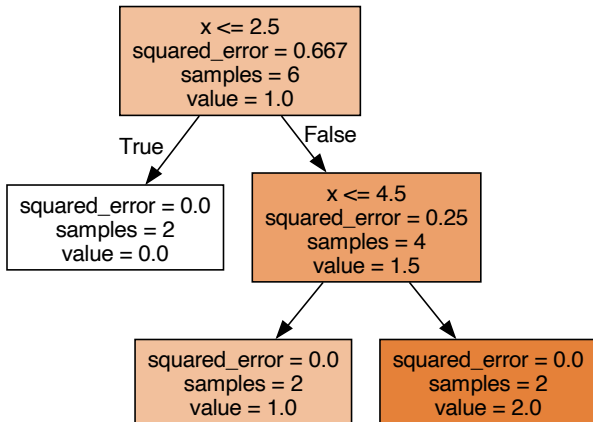
Notebook: [decision-tree-real-input-real-output.html](#)



# Example 1

## The Decision Boundary

Notebook: [decision-tree-real-input-real-output.html](#)



# Objective Function for Regression Trees

Feature is denoted by  $X$  and target by  $Y$ .

Let the split be at  $X = s$ .

Define regions:  $R_1 = \{x : x \leq s\}$  and  $R_2 = \{x : x > s\}$ .

# Objective Function for Regression Trees

Feature is denoted by  $X$  and target by  $Y$ .

Let the split be at  $X = s$ .

Define regions:  $R_1 = \{x : x \leq s\}$  and  $R_2 = \{x : x > s\}$ .

For each region, compute the mean prediction:

$$c_1 = \frac{1}{|R_1|} \sum_{x_i \in R_1} y_i$$

$$c_2 = \frac{1}{|R_2|} \sum_{x_i \in R_2} y_i$$

## Objective Function for Regression Trees

Feature is denoted by  $X$  and target by  $Y$ .

Let the split be at  $X = s$ .

Define regions:  $R_1 = \{x : x \leq s\}$  and  $R_2 = \{x : x > s\}$ .

For each region, compute the mean prediction:

$$c_1 = \frac{1}{|R_1|} \sum_{x_i \in R_1} y_i$$

$$c_2 = \frac{1}{|R_2|} \sum_{x_i \in R_2} y_i$$

The loss function is:

$$\text{Loss}(s) = \sum_{x_i \in R_1} (y_i - c_1)^2 + \sum_{x_i \in R_2} (y_i - c_2)^2$$



# Objective Function for Regression Trees

Feature is denoted by  $X$  and target by  $Y$ .

Let the split be at  $X = s$ .

Define regions:  $R_1 = \{x : x \leq s\}$  and  $R_2 = \{x : x > s\}$ .

For each region, compute the mean prediction:

$$c_1 = \frac{1}{|R_1|} \sum_{x_i \in R_1} y_i$$

$$c_2 = \frac{1}{|R_2|} \sum_{x_i \in R_2} y_i$$

The loss function is:

$$\text{Loss}(s) = \sum_{x_i \in R_1} (y_i - c_1)^2 + \sum_{x_i \in R_2} (y_i - c_2)^2$$

Our objective is to find the optimal split:

## Algorithm: Finding the Optimal Split

1. Sort all data points  $(x_i, y_i)$  in increasing order of  $x_i$ .

## Algorithm: Finding the Optimal Split

1. Sort all data points  $(x_i, y_i)$  in increasing order of  $x_i$ .
2. Evaluate the loss function for all candidate splits:

$$s = \frac{x_i + x_{i+1}}{2} \text{ for } i = 1, 2, \dots, n - 1$$

## Algorithm: Finding the Optimal Split

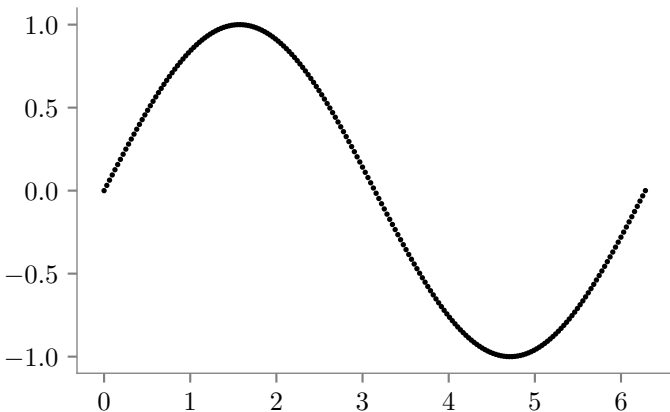
## A Question!

Draw a regression tree for  $Y = \sin(X)$ ,  $0 \leq X \leq 2\pi$

## A Question!

Dataset of  $Y = \sin(X)$ ,  $0 \leq X \leq 7$  with 10,000 points

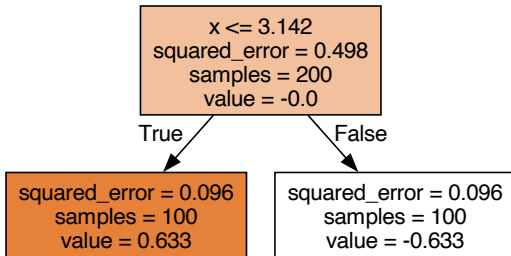
Notebook: [decision-tree-real-input-real-output.html](#)



# A Question!

Regression tree of depth 1

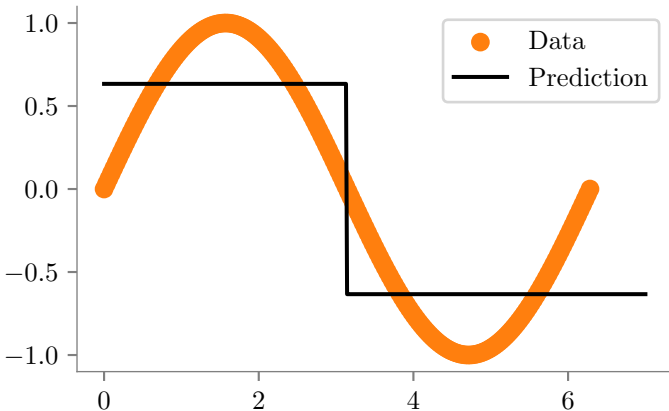
Notebook: [decision-tree-real-input-real-output.html](#)



# A Question!

## Decision Boundary

Notebook: [decision-tree-real-input-real-output.html](#)

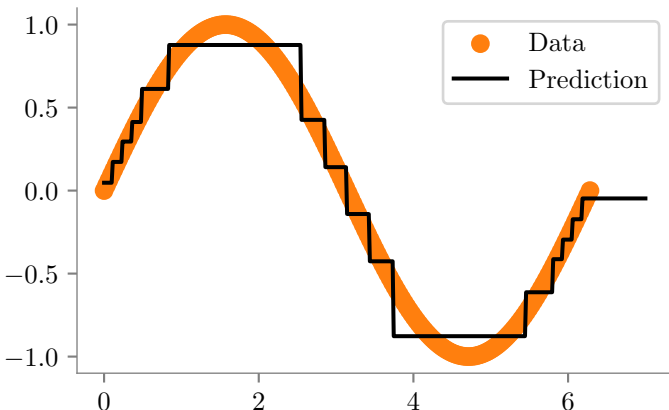




## A Question!

Regression tree with no depth limit is too big to fit in a slide. It has of depth 4. The decision boundaries are in figure below.

Notebook: [decision-tree-real-input-real-output.html](http://decision-tree-real-input-real-output.html)



# Summary

- Interpretability an important goal

# Summary

- Interpretability an important goal
- Decision trees: well known interpretable models

# Summary

- Interpretability an important goal
- Decision trees: well known interpretable models
- Learning optimal tree is hard

# Summary

- Interpretability an important goal
- Decision trees: well known interpretable models
- Learning optimal tree is hard
- Greedy approach:

# Summary

- Interpretability an important goal
- Decision trees: well known interpretable models
- Learning optimal tree is hard
- Greedy approach:
- Recursively split to maximize “performance gain”

# Summary

- Interpretability an important goal
- Decision trees: well known interpretable models
- Learning optimal tree is hard
- Greedy approach:
- Recursively split to maximize “performance gain”
- Issues:

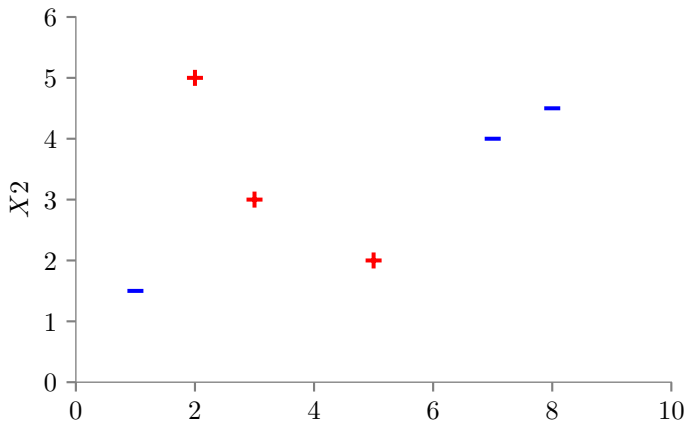
# Summary

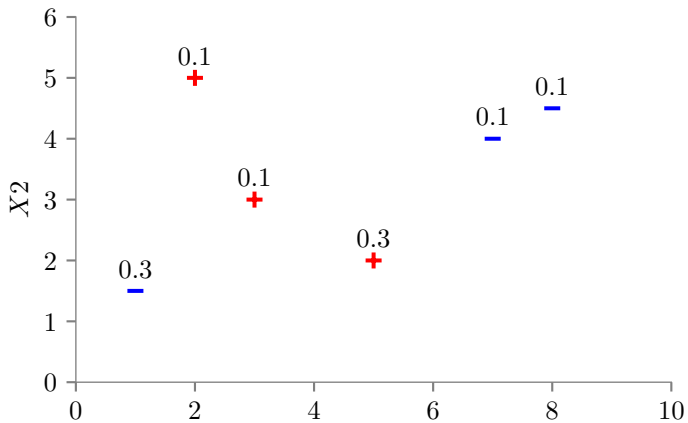
- Interpretability an important goal
- Decision trees: well known interpretable models
- Learning optimal tree is hard
- Greedy approach:
- Recursively split to maximize “performance gain”
- Issues:
  - Can overfit easily!

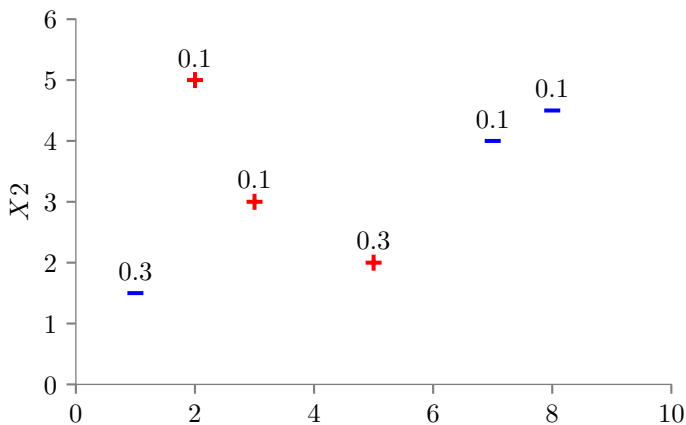


# Summary

- Interpretability an important goal
- Decision trees: well known interpretable models
- Learning optimal tree is hard
- Greedy approach:
- Recursively split to maximize “performance gain”
- Issues:
  - Can overfit easily!
  - Empirically not as powerful as other methods





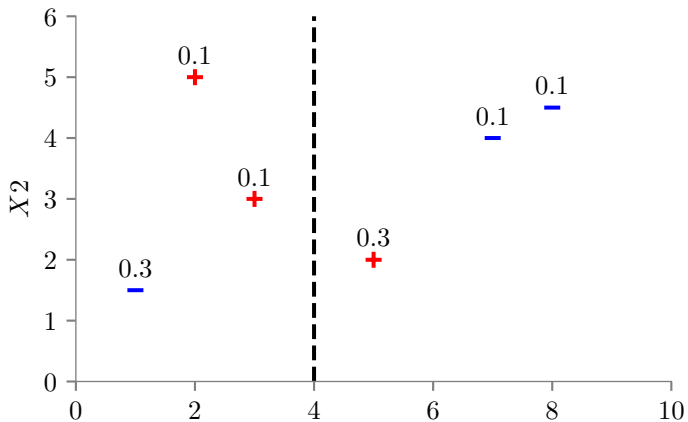


$$= -P(+)\log_2 P(+)-P(-)\log_2 P(-)$$

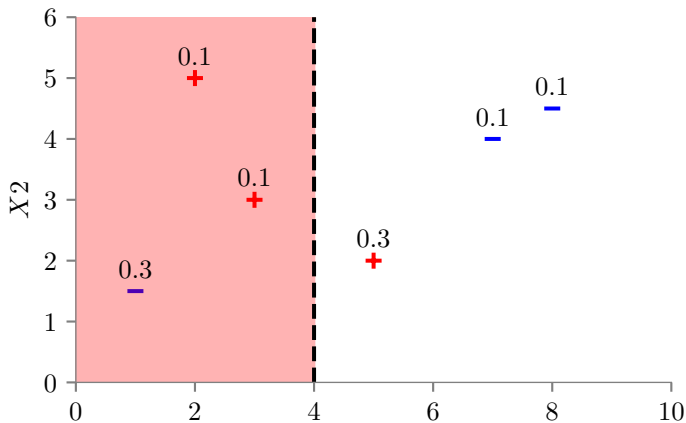
$$P(+)=\frac{0.1+0.1+0.3}{1}=0.5,\quad P(-)=\frac{0.3+0.1+0.1}{1}=0.5$$

# Weighted Entropy

---

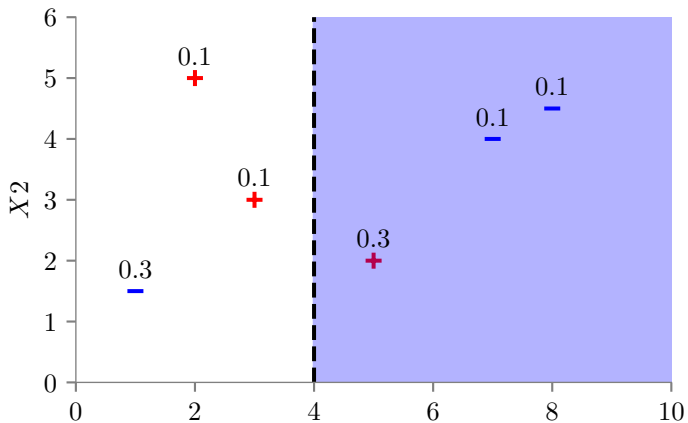


Candidate Line:  $X_1 = 4(X_1^*)$



Entropy of  $X1 \leq X1^* = E_{S(X1 < X1^*)}$

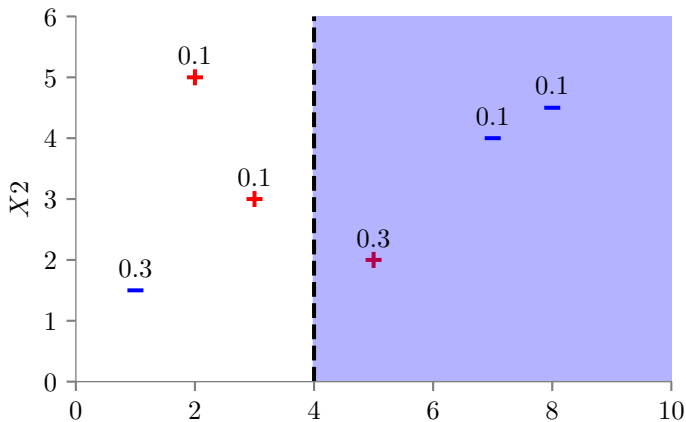
$$P(+)=\frac{0.1+0.1}{0.1+0.1+0.3}=\frac{2}{5}$$



Entropy of  $X_1 > X_1^* = E_{S(X_1 > X_1^*)}$

$$P(+)=\frac{3}{5}$$





$$IG(X_1 = X_1^*) = E_S - \frac{0.5}{1} \cdot E_{S(X_1 < X_1^*)} - \frac{0.5}{1} \cdot E_{S(X_1 > X_1^*)}$$